

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

Reference 1.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-267938

(43)Date of publication of application : 28.09.2001

(51)Int.Cl.

H03M 13/45
G06F 11/10

(21)Application number : 2001-024226

(71)Applicant : TEXAS INSTR INC <TI>

(22)Date of filing : 31.01.2001

(72)Inventor : GATHERER ALAN
WOLF TOD D
LAINE ARMELLE

(30)Priority

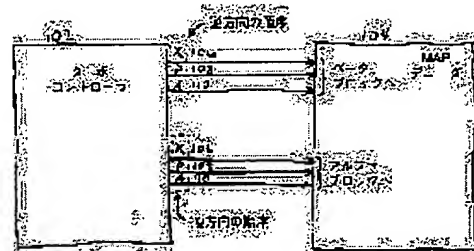
Priority number : 2000 179055 Priority date : 31.01.2000 Priority country : US

(54) MAP DECODING USING PARALLEL-PROCESSED SLIDING WINDOW PROCESSING

(57)Abstract:

PROBLEM TO BE SOLVED: To conduct MAP(maximum a posteriori) decoding by means of a pipeline window processing.

SOLUTION: This invention provides a sub-block processing technology in the MAP decoding employing the pipeline processing. The alpha processing is started in parallel with the beta processing and each stage of the processing sets is internally processed in parallel. Applying pipeline processing to the forward propagation processing and the backward propagation processing can ingeniously enhance the throughput to some degrees.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-267938

(P2001-267938A)

(43) 公開日 平成13年9月28日 (2001.9.28)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 3 M 13/45		H 0 3 M 13/45	
G 0 6 F 11/10	3 3 0	G 0 6 F 11/10	3 3 0 S

審査請求 未請求 請求項の数 2 O L (全 10 頁)

(21) 出願番号 特願2001-24226 (P2001-24226)

(22) 出願日 平成13年1月31日 (2001.1.31)

(31) 優先権主張番号 1 7 9 0 5 5

(32) 優先日 平成12年1月31日 (2000.1.31)

(33) 優先権主張国 米国 (U S)

(71) 出願人 590000879

テキサス インストルメンツ インコーポ
レイテッド

アメリカ合衆国テキサス州ダラス、ノース
セントラルエクスプレスウェイ 13500

(72) 発明者 アラン ギャザー

アメリカ合衆国 テキサス、リチャードソ
ン、ブルー ボネット ドライブ 2105

(72) 発明者 トッド ディ、ウォルフ

アメリカ合衆国 テキサス、リチャードソ
ン、オーク ブルック ドライブ 2107

(74) 代理人 100066692

弁理士 浅村 皓 (外3名)

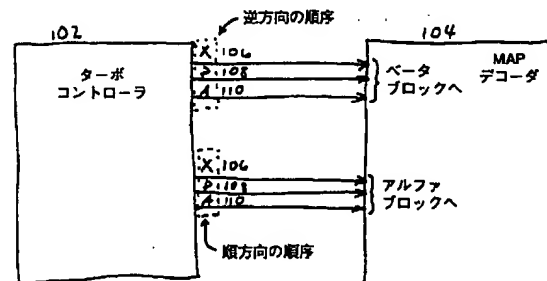
最終頁に続く

(54) 【発明の名称】 パラレル化されたスライディングウィンドウ処理によるMAPデコーディング

(57) 【要約】

【課題】 パイプライン状のウィンドウ化された処理によりMAPデコーディングを行うこと

【解決手段】 パイプライン化を使用するMAPデコーディングにおけるサブブロック処理技術に関する。ベータの処理とパラレルにアルファの処理を開始し、処理の各ステージを更に内部でパラレルにする。前方伝搬処理と後方伝搬処理とをパイプライン化することによって、更にある程度スループットを純粋に改善できる。



【特許請求の範囲】

【請求項 1】 第 1 誘導パラメータを得るように、データの少なくとも部分的ブロックに第 1 方向の第 1 のスライディングウィンドウ演算を実行する工程と、

第 2 の誘導パラメータを得るように、前記データの少なくとも一部のブロックに、前記第 1 の方向と反対の第 2 の方向の第 2 のスライディングウィンドウ演算を実行する工程と、

データ推定値を発生するように、前記第 1 の誘導パラメータおよび前記第 2 の誘導パラメータを処理する工程とを備え、データのそれぞれの異なる部分に対し平行に演算を行うように、前記スライディングウィンドウ演算を互いにパイプライン化する、MAP デコーディング方法。

【請求項 2】 スライディングウィンドウブロックに分割されたデータストリームを MAP 処理するためのシステムであって、

アルファ発生プロセスと、

ベータ発生プロセスとを備え、

アルファプロローグおよびベータプロローグを使用して多数のスライディングウィンドウブロックに対する演算を行うように、アルファ発生プロセスとベータ発生プロセスとが多数のパイプライン化ステージに分割されている、MAP 処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は無線通信技術に関し、より詳細にはターボデコーディングおよび同様な技術に関する。

【0002】

【従来技術】 誤り訂正

符号化されたデジタル通信システムは所定の信号対ノイズ比 (SNR) におけるデータの信頼性を改善するために、誤り制御用符号を使用している。例えば (データ記憶アプリケーションで使用される) 極端に簡単な形態は、8 ビットのデータごとに 1 つのパリティビットを発生し、送信する方法である。9 ビットの各ブロックごとにパリティをチェックすることにより、単一ビットの誤りを検出できる。(各ブロックに 3 つの誤り訂正ビットを加えることにより、単一ビットの誤りを検出し、訂正することができる。) 一般に誤り制御符号化技術としてデータストリームに付随するようにエクストラビットを発生し、データストリーム内の誤りを検出し、可能な場合にはこの誤りを訂正できるようにする技術が多数存在している。

【0003】 トレリス符号化

誤り制御のための重要な技術の 1 つとしてトレリス符号化方法がある。この種の技術では所定のシンボルの次に別のシンボルが直接続くことができないように、シンボルのシーケンスにある制限条件が課される。これら制限

条件は許容される変化と許容されない変化との幾何学的パターン (すなわちトレリス) によって定められることが多い。シンボルのシーケンスに課される制限条件が存在することによりデータシーケンスに対するある構造が得られる。すなわち制限条件が破られたかどうかを分析することにより多数の誤りを訂正できる。この方法は極めて強力な種類の符号化技術であり、これら幾何学的制限条件をより高い次元としたり、または制限条件を表記するために代数式を使用することもでき、多数の変形例を使用できる。

【0004】 ターボ符号化

ターボ符号化アーキテクチャのエンコーダ側では一般に 2 つのエンコーダを使用する。すなわち 1 つのエンコーダは未処理のデータストリームに演算を実行し、他方のエンコーダはベースデータストリームのシャッフルされたコピーに演算を実行し、未処理のデータストリームの各ビットに対し 2 つのパリティビットを発生するようになっている。従って、エンコーダの出力は入進データストリームのビットの 3 倍多いビットを含む。以下、このような「パラレル接続エンコーダ」(すなわち PCE) のコンフィギュレーションについて詳細に説明する。

【0005】

【発明が解決しようとする課題】 このターボ符号化の最も驚くべきことは、そのデコーディングのアーキテクチャにある。デコーダ側ではオリジナルデータを再生するために (もしチャンネルにノイズがなかった場合) エンコーダ側で行われた変換を単に反転するプロセスを呼び出す。しかしながら、デコーダ側は情報ビットのソフト推定値に演算を行うように構成されており、繰り返し再推定プロセスにより推定値を正確にする。デコーダは第 1 回のパスで判断に到達する必要はないが、一般に収束が得られるまで情報ビットの推定値を繰り返して改善することが認められている。

【0006】 MAP デコーダ MAP デコーディングは計算を集中させる技術であり、この技術はターボデコーディングおよびトレリス符号化変調技術にとって極めて重要なものとなっている。MAP とは「帰納的後 (maximum a posteriori)」を示すものである。MAP デコーダは初期に受信したシンボルおよび最近受信したシンボルを検討し、各シンボルに対する最も可能性の高い推定値を必要とする。このことは、トレリス符号化を使用する場合に特に重要である。その理由は、各シンボルに対する推定値はその後続くシンボルに対する推定値と関連しているからである。

【0007】 これと対照的に、最尤 (ML) デコーダは実際に受信したシーケンスの確率が最も高くなる送信シーケンスを計算しようとする。これらの言葉による説明は同様に聞こえるが、MAP デコーディングと ML デコーディングとの差は極めて大きい。ML デコーディングのほうが計算上簡単であるが、多くのアプリケーション

3

ではMAPデコーディングが必要とされている。

【0008】MAPデコーディングは前方伝搬推定値と後方伝搬推定値とを通常組み合わせている。すなわち、あるシーケンスの受信したシンボルを記憶し、次にあるシーケンスの前方遷移確率を発生するように一方向に

(すなわ時間的に前方に)処理し、次にあるシーケンスの後方遷移確率を発生するように逆方向に(時間的に後方に)処理を行う。前方および後方遷移確率と実際に受信した信号に対するデータとを組み合わせることによって、各シンボルに対する純推定値を発生する(この方法の更なる細部については、パール、コック、ジェリネックおよびラパイブによる論文、「シンボル誤り率を最小にするためのリニア符号のデコーディング」(情報理論に関するIEEEトランザクション、1974年)を参照されたい。本書ではこの論文を参考例として援用する)。

【0009】前方計算と後方計算とを組み合わせるにはかなりの量のメモリが必要である。進歩したセルラー通信におけるブロックは大きく(すなわち5120個のシンボル)なり得るので、ブロックの各シンボルに対する可能な各遷移に対する値を記憶するのに必要なメモリは大きくなる。デコーディング中のメモリ条件を減らすために、データの各ブロックをMAPデコーディング用のより小さい多数のブロック(例えば128シンボルの40ブロック)に分割できる。

【0010】トレリス符号化はデータの完全なブロックに対して行われるので、完全ブロックに対しスタート状態および終了状態は既知となっている。しかしながら、中間部に対してスタート状態と終了状態とは既知ではない。これによってこのようなより小さいブロックの正確なプロセスでは問題が生じるが、各ブロックをスタートする前に数個のシンボルに対し、前方推定プロセスを繰り返すだけで、ブロック内の第1シンボルの処理を初期値の良好な組からスタートすることを保証できる。

【0011】

【課題を解決するための手段】パイプライン状のウィンドウ化された処理によるMAPデコーディング

本願はパイプライン化を使用するMAPデコーディングにおけるサブブロック処理技術を開示するものである。ベータの処理と平行にアルファの処理を開始する。処理の各ステージを更に内部で平行にすることが好ましいが、前方伝搬処理と後方伝搬処理とをパイプライン化することによって、更にある程度スループットを純粹に改善できる。

【0012】種々の実施例における本書に開示する方法および構造の利点としては次のことの1つ以上が含まれる。

一処理をより高速にできること

一メモリをより少なくできること

4

ターボデコーダにおける可能な繰返し回数をより多くできること

【0013】本発明の重要なサンプル実施例を示し、参考例として本明細書の一部である添付図面を参照しながら、本書に開示する発明について説明する。

【0014】

【発明の実施の形態】以下、現時点で好ましい実施例を特に参照し、本願の多数の革新的要旨について説明する。しかしながら、この種の実施例は革新的要旨の多数の有利な使用法の数例しか示していないと理解すべきである。一般に、本願明細書における説明は請求した種々の発明を必ずしも限定するものではない。更に一部の説明は発明の特徴に適用できるが、他の特徴には適用できない。

【0015】システムハードウェアを同時に作動させることにより、2つ以上の基本的動作を同時に処理することが可能である。2つの周知の技術、すなわちパラレル化およびパイプライン化によって同時処理を実現することが多い。

【0016】パラレル化はシステムにおけるハードウェア構造を複製することを含む。解決すべき問題の異なる部分で多数の構造を同時に実施させることによって性能を改善できる。

【0017】パイプライン化により実行すべき機能小さい部分に分割し、各部分に別個のハードウェアを割り当てる。パラレル化およびパイプライン化に関する多くの情報は、コググ著「パイプライン化コンピュータのアーキテクチャ」に記載されており、本明細書ではこの文献を参考例として援用する。

【0018】図1はターボデコーダのブロック図を示す。ここには2つの主要ブロック、すなわちターボコントローラ102とMAPデコーダ104とが示されている。ターボコントローラ102はMAPデコーダ104に対する入力信号として働くデータストリーム(X、システムティックなデータ106;P、パリティデータ108;およびA、アプリアリ(A PRIORI)なデータ110)を記憶し、MAPデコーダ104にデータを入力する順序を制御する。図は、3つのデータストリームの各々がMAPデコーダ104に2回入力されることを示す。アルファおよびベータ発生ブロックは逆方向の順序でデータを入力するのを求めるので、2つの別個の組の入力データが必要である。MAPデコーダ104の外因性出力データは別のデコーダの繰返しのためにコンピュータ102へ戻される。

【0019】図2は、パラレルなスライディングウィンドウ処理を使用するMAPデコーダ104のブロック図を示す。MAPデコーダ104はその入力信号としてスケーリングされたシステムティックなデータ信号106と、スケーリングされたパリティデータ信号108と、アプリアリ信号110とを受信する。X信号106はN

個あり、ここでNはインターリーバーのサイズである。各ベータステートベクトル208およびアルファステートベクトル210に対し、N個のX信号106が加えられ、これらベクトルはそれぞれベータブロック202およびアルファブロック206の出力信号である。ベータの発生中にX106は逆方向の順序で印加され、アルファの発生中はX106は順方向の順序で印加される。P信号108もN個存在する。各アルファベクトル210およびベータベクトル208に対し、N個のP信号108が印加される。ベータの発生中はP108は逆方向の順序で印加され、アルファの発生中は順方向の順序で印加される。アプリアリ110は先のMAPデコード動作からのインターリーブされた外因性データまたはディンターリーブされた外因性データのいずれかである。このアプリアリ信号110はN個あり、各ベータベクトル208およびアルファベクトル210に対し1つのアプリアリ信号110が印加される。ベータおよびアルファの発生のためにX信号106およびP信号108と同じ方向にアプリアリ信号110が印加される。

【0020】図3により詳細に示されたベータ発生部分202は入力信号X106、P108およびA110を受信し、ベータステートベクトル208を発生し、このベクトルはベータRAM204に記憶される。アルファ発生部分206は（ベータ入力信号に対し逆方向の順序で）入力信号X106、P108およびA110を受信する。図4により詳細に示されたアルファ発生ブロック206はアルファステートベクトル210を発生する。アルファ発生部分の出力208およびベータ発生部分の出力210は図5に示された外因性データ発生部分212に対する入力信号として働く。これらデータストリームは外因性データ発生部分212に投入される前にパリティストリームP108により正しくシーケンス制御しなければならない。

【0021】図3は、ベータ発生ステージを示す。MAPリセットステート中、最初にレジスタはベータステートベクトル208のための初期状態にセットされる。データを符号化するのに使用されるトレリスに従い、加算ツリー302によりベータ信号208とX106とP108とA110とが共に加算する（好ましい実施例では8ステートトレリスが使用される。）この結果はレジスタ310に記憶される。第2ステージでは加算器302の結果は、8MAX*ブロック304へ印加され、次にMAX*レジスタ312へ記憶される。次に正規化されていない出力信号が2つの別個の正規化ステージ306、308へ進む。これらステージの各々は結果を記憶するためのレジスタ314、316を有する。従って、全体のプロセスは完了するまでに4つのクロックサイクルを必要とするベータ発生ブロック202のフィードバックループ内に4つのステージを有する。このレイテンシー（4つのクロックサイクル）は利用できるパイプ

イン化のレベルを決定する。

【0022】図4にはアルファ発生部分206が示されている。第1に、レジスタは初期条件にセットされ、次に加算器402によりデータ入力信号が加算され、結果がレジスタ412に記憶される。次に、これら結果はMAX*ブロック406へ入力され、MAX*レジスタ414に記憶される。このアルファ発生部分206も2つの正規化ステージ408、410を有し、各ステージは自己のレジスタ416、418を有する。従って、アルファ発生ステージ206のレイテンシーは4であり、4つのレベルのパイプライン化を実現できる。

【0023】図5に示された外因性データ発生部分212はアルファ発生部分とパラレルに作動する。使用するトレリスに従い、加算器502によりアルファ210とベータ208とP108とが加算され、これら結果はレジスタ510に記憶される。第2ステージではこれら結果はMAX*ブロック504へ印加され、MAX*レジスタ512に記憶される。これらの結果は再度MAX*ブロック504に印加され、次にレジスタ508に記憶される。この結果は加算され、別のレジスタ514に記憶され、出力信号は外因性信号214となる。

【0024】スライディングウィンドウによるパラレル化

スライディングウィンドウ方法は基本的には入進データのN個のサイズのブロックをより小さい数個のブロックに分割することから成る。これら、より小さいブロックの各々はスライディングウィンドウブロックと称される。これらスライディングウィンドウブロックは各々別々にMAPデコード化され、アルファベクトルおよびベータベクトルの双方に対するプロローグを有する。個々のアルファスライディングウィンドウブロックおよびベータスライディングウィンドウブロックに対するデコード化はパラレルに実行される。個々のスライディングウィンドウブロックに対して初期条件は既知ではないので、初期値の良好な組に達するのにプロローグが使用される。

【0025】先のブロックの十分内部にあるポイントにおいて、アルファの更新をスタートすると共に、次のブロックの十分内部にあるポイントでベータの更新をスタートすることにより、デコードは初期条件を忘れることができ、実際のデータに対する演算を開始する前に収束することができる。使用されるプロローグ部分のサイズはトレリス内のステートの数のほぼ3または4倍である。最初のアルファスライディングブロックおよび最終ベータスライディングブロックは既知のステートから発生し、それぞれのプロローグ部分のサイズは（例えば）8ステートトレリスに対して3となる。

【0026】革新的なアルファプロローグによってデータのアルファスライディングウィンドウブロックおよびベータスライディングウィンドウブロックの双方のパラ

7

レル処理が可能となる。使用する特定の実現例に応じ、アルファまたはベータの各更新には実行に数クロックサイクル（上記実施例では4クロックサイクル）かかる。このレイテンシーはシステムで可能なパイプライン化の程度を決定する。好ましい実施例では、各アルファブロックおよびベータブロック内でのパイプライン化には4つのレベルがある（このことは、アルファ発生ステージおよびベータ発生ステージの各々におけるデータがパイプライン化されること、すなわち別個の組みのデータに分割され、ベータ発生部分における連続するステージによって別々に演算されることを意味する）。アルファブ

10 ック自身とベータブロック自身の間にはある程度の平行性もある。このことは2つの部分が外因性入力信号を発生するように同時に作動することを意味する。
【0027】アルファベクトル発生方法およびベータベクトル発生方法は上に示すように多数のステージに分割される。これらステージは図3および4に示されたアルファおよびベータベクトル発生の繰り返ループ内にある。ステージの数は特定のアーキテクチャに対するレイ

20 テンシーと同じとなる。好ましい実施例では、これらステージは加算器、MAX*および2つの正規化ステージである。これらステージのレイテンシーは可能な平行処理の程度を決定する。例えば好ましい実施例ではこのレイテンシーは4であり、このことは4つのスライディングウィンドウブロックを平行に処理できることを意味する。従って、4つのスライディングウィンドウブロックで1つのサブブロックを構成する。
【0028】図6にはスライディングブロックのパイプ

30 ライン化が示されている。最初のクロックサイクル中に、ベータ0（第1スライディングブロック）が加算器ステージに進入する。第2クロックサイクルで、ベータ0がMAX*ステージへ進入し、ベータ1が加算器ステージに進入する。第3クロックサイクルではベータ0が第1正規化ステージ（ベータ発生の第3ステージ）に進入し、ベータ1がMAX*ステージに進入し、ベータ2

40 が加算器ステージに進入する。次にベータ0が第2正規化ステージに進入し、ベータ1が第1正規化ステージに進入し、ベータ2がMAX*ステージに進入し、ベータ3が加算器ステージに進入する。各ステージのための中間値が上記に示すようなレジスタに記憶される。
【0029】外因性部分に入力されるデータを同期化で

8

のスライディングブロックである）を処理できる。次に、アルファ部分のスライディングブロックの第1の組を処理しながら、ベータ部分のスライディングブロックの第2の組（すなわち第2のサブブロック）を処理する。アルファ部分と平行に外因性部分を処理する。これによりアルファステート部分およびベータステートベクトルの双方を記憶するためのメモリ条件が低減される。その理由は、アルファ出力が発生した際にこれらを外因性部分に直接印加できるからである。外因性部分は一度に1つの出力サブブロックを発生する（更に入力信号を必要とする）ので、ベータRAMは一度に1つのサブブロックを記憶するだけでよい。（アルファ処理とベータ処理とを反転できることに留意されたい。）このようにするにはアルファ出力をメモリに記憶しなければならず、ベータブロックと外因性部分を平行に作動する。

【0030】図8はアルファスライディングウィンドウブロックとベータスライディングウィンドウブロックとの間の対応を示す。全データブロックはN個のシンボル + 多数のテールビットとから成る。このブロックはサブブロックに分割され、更にサブブロックはスライディングウィンドウブロックに分割される。1クロックサイクルごとに1つのスライディングウィンドウブロックが処理される。各スライディングウィンドウブロックは1つのプロローグを含み、ベータプロローグはスライディングウィンドウの右までの数個のシンボルから成り、アルファプロローグはスライディングウィンドウの左までの数個のビットから成る。このことは、図における連続するスライディングブロック間が重なっていることによって示されている。各ベータスライディングウィンドウはアルファスライディングブロックに対して逆方向に処理される。

【0031】図9は、ベータビットおよびアルファビットを処理する順序の一例を示す。この例はスライディングウィンドウのサイズを100とし、プロローグ長さを24とし、1サブブロックごとに4つのスライディングウィンドウがあると仮定している。スライディングブロックのベータ0はビット123におけるプロローグのスタート点で開始し、次にプロローグはビット100で終了する。信頼性データはビット99で開始し、ビット0で終了する。アルファスライディングブロックも同様に分割されている。（スタートポイントおよび終了ポイントは既知であるので、ブロックの開始に対するプロローグがないために、アルファスライディングブロックに対する最初の2つのエントリは存在しないことに留意されたい。）

【0032】外因性データはアルファ発生プロセスおよびベータ発生プロセスの双方と平行に処理できない。その理由は、アルファ、ベータおよびパリティデータからのデータを必要とする外因性入力データは所定の

順序で入力しなければならないからである。次の説明は外因性入力のインデクシングを示す。(アルファ0およびベータ0に対応する)E0はビット0から99まで進む。スライディングウィンドウのサイズを100とした場合、E1は100~199まで進み、次々に同じように進む。この例が必要とする入力は次のようになる。最初のクロックサイクルではアルファ、ベータおよびPからのビット0に関連するソフト推定データが外因性データに入力される。第2クロックサイクルでは3つの入力からのビット100に関連するデータが必要とされる。第3クロックサイクルでは、ビット200に関連するデータが必要とされ、第4クロックサイクルでは、ビット300に関連するデータが必要とされ、第5クロックサイクルでは、入力はビット1に関連するデータに戻る(最初のクロックサイクルの入力は1ビットだけシフトされる)。次のサイクルではビット101のデータが必要とされ、次々に同様なデータが必要とされる。従って、ベータビットが発生されるとこれらベータビットをRAMに記憶しなければならない。その理由は、これらベータビットはアルファビットおよびパリティビットと異なる順序で発生され、アルファビットおよびパリティビットの発生時には必要でないからである。対応するアルファおよびベータが発生されると、外因性データを計算できる。

【0033】定義

次は、本願で使用される技術用語の一部の通常の意味の短い定義である。(しかしながら当業者であればコンテキストが異なる意味を必要とするかどうかはわかるであろう。)標準的な技術的辞書および雑誌には別の定義を見い出すことができる。

【0034】MAX*: このMAX*は次の式で示される自然対数関数に対する最大値を見出す近似式である。

【0035】

【数1】

$$\ln[e^A + e^B] \approx \text{MAX}^* = \text{MAX}(A+B) + f(|A-B|)$$

【0036】ここで、 $f(A-B)$ は補正項である。上記式を近似させるこの値に対しては、通常、ルックアップテーブルが使用される。

【0037】

【数2】

$$\ln[1 + e^{A-B}]$$

【0038】ルックアップテーブルの代わりに上記式を使用する場合、MAX*の定義は近似値ではなく、正確な等しい値となる。

【0039】MAPデコーダ: 帰納的的最大MAPデコーダは受信情報をrとした場合、シンボルxの確率 $p(x/r)$ を最大にするxを選択する検出基準を使用する。外因性データ: デコードされるビットの値を推定するデ

コードの出力である。この外因性データは通常ソフト推定値である。

【0040】変形および変更例: 当業者であれば認識できるように、本願に説明した革新的原理は広範な応用例にわたって変形および変更できるので、特許請求の範囲に記載した要旨の範囲はこれまで示した特定の例のいずれかによって変形されるものではなく、発行される特許請求の範囲によってしか定義されない。

【0041】好ましい実施例が特に詳細に説明されているが、本書に開示した発明の要旨の範囲から逸脱することなく、発明を実施するに際し、多くの変形が可能である。例えば(レジスタもしくは他の手段を追加することにより)各ステートベクトル発生ステージのレイテンシーを変えることができるので、可能なパイプライン化の程度も変わる。更に実施例で適用された発明の要旨を変えることなくトレリスのサイズを変えることもできる。必要なRAM記憶装置をわずかに変えるだけで、ベータ、アルファおよび外因性データを種々のパラレルな組み合わせで発生することができる。

【0042】次に、当業者であれば、広範な範囲のアプリケーションをカバーするように、本願で使用される入力信号の定義(システムティックデータXおよびパリティデータP)を一般化できると認識できよう。例えばこれら入力信号はMAP等化もしくはターボトレリスコーディングのようなアプリケーションで変わり得る。一部のアプリケーションでは入力信号をビットのソフト推定値とすることはできないが、むしろ他の変数のソフト推定値とすることができる。個々に開示する革新的技術は、かかる実現における変形例のすべてをカバーするものである。

【0043】本願の開示された革新技術はどのMAPアーキテクチャにも適用できる。例えばMAPデコーダを使用するターボデコーダにおける、ここに開示した発明の要旨の実現は、本発明の範囲内のものである。任意のMAP演算、例えばMAP等化も本願の範囲内のものであり、MAP等化とは観察された出力信号を発生するためにトレリスで制限されたチャンネルへ入力されるデータとしてチャンネル関数を記述するプロセスである。次に、トレリス図におよび観察されたチャンネル出力にMAPデコーダを適用することにより、最大アプリアリな意味で(in a maximum a priori sense)、チャンネルへの入力信号を推定できる。これは、(a)等化器からのソフト出力が必要である場合、(b)リニアフィルタまたは等化器を使って得られるよりも、チャンネルへの入力信号のより正確な推定値が必要な場合、もしくは(c)チャンネルおよび適用されるFECの繰返しジョイントデコードが必要な場合に、有効となる。一般に、MAPは観察されたデータがリアトレリスへの入力信号によって発生されたと判った状況で使用できる。

【0044】同様に、ソフトウェアだけでなくハードウェアを使ったMAPアーキテクチャも本発明の範囲内にある。今日のDSPでは、データバスの徹底的なパイプライン化を用いることによって極めて高い処理レートを達成している。このことは、ベータおよびアルファの更新のようなフィードバックプロセスではDSPを効率的に使用できないことを意味する。本発明を使用することにより、DSPによっていくつかのブロックを同時にパイプライン状に処理することが可能となっており、このことによって徹底的にパイプライン状となったDSPアーキテクチャで演算をかなりスピードアップできる。

【0045】ハガードおよびヴィッカー著「ターボコーディング」、シュレーゲル著「トレリスコーディング」、ヴィッカー著「誤り制御システムズ」およびアンドリュウ・ピタルピ著「たたみ込み符号のためのMAPデコーダの集中的正当化および簡略化された実現」、通信の選択された領域に関するIEEEジャーナル、第16巻第2号(1998年2月)には、MAPデコーダおよびコーディングにおける技術状態に関する別の背景資料を見いだすことができる。これら文献のすべてを本書で参考例として援用する。

【0046】以上の説明に関して更に以下の項を開示する。

(1) 第1誘導パラメータを得るように、データの少なくとも部分的ブロックに第1方向の第1のスライディングウィンドウ演算を実行する工程と、第2の誘導パラメータを得るように、前記データの少なくとも一部のブロックに、前記第1の方向と反対の第2の方向の第2のスライディングウィンドウ演算を実行する工程と、データ推定値を発生するように、前記第1の誘導パラメータおよび前記第2の誘導パラメータを処理する工程とを備え、データのそれぞれの異なる部分に対し平行に演算を行うように、前記スライディングウィンドウ演算を互いにパイプライン化する、MAPデコーディング方法。

【0047】(2) 前記スライディングウィンドウ演算の各々を別個のステージに分割し、これら別個のステージがデータの異なる部分的ブロックに対し平行に演算を行う、第1項記載の方法。

【0048】(3) 少なくとも1つのシーケンス制限条件に従い、終了ポイントに必ずしも既知のステートを有しないデータのブロックを双方向に処理するための方法であって、前記シーケンス制限条件に従った前記第1の方向のプロローグ要素の第1の処理の後に、第1方向にブロックのデータ要素をシーケンシャルに処理する工程と、前記シーケンス制限条件に従った前記第2の方向のプロローグ要素の第1の処理の後に、第2の方向に前記データ要素をシーケンシャルに処理する工程とを備えた、データのブロックを双方向に処理する方法。

【0049】(4) 第1方向のデータ要素の処理と第2

方向のデータ要素の処理とを平行に実行する、第3項記載の方法。

【0050】(5) データの要素を処理する各工程を別個のステージに分割し、別個のステージが異なるデータ要素に対し平行に演算を行う、第3項記載の方法。

【0051】(6) データのラティス符号化ブロックを平行MAP処理する方法であって、データをスライディングウィンドウブロックに分割する工程を含み、前記スライディングウィンドウブロックの多数のブロックの各々に対し、

a) シーケンス制限条件に従った前記第1方向のプロローグ要素の第1処理の後に、第1方向にそれぞれのスライディングウィンドウブロックの要素をシーケンシャルに処理する工程と、

b) シーケンス制限条件に従った前記第2方向のプロローグ要素の第1処理の後に、第2方向にそれぞれのスライディングウィンドウブロックの要素をシーケンシャルに処理する工程とを実行し、前記工程a)およびb)を互いに少なくとも部分的に平行に実行する、平行MAP処理方法。

【0052】(7) 工程a)および/またはb)を別個のステージに分割し、これら別個のステージが異なるステージウィンドウブロックに対し平行に演算を行う、第6項記載の方法。

【0053】(8) a) 少なくとも1つの加算ツリーにおいて、確率メトリクスを組み合わせる工程と、

b) 別の可能性に対応する前記メトリクスの1つずつを組み合わせるように、最大値を見出す演算を実行する工程とを備え、前記工程a)とb)とを互いに平行化されたパイプライン状態で少なくとも部分的に実行する、平行MAP処理方法。

【0054】(9) 最大値を見出す演算が指数-対数式である、第8項記載の方法。

【0055】(10) 最大値を見出す演算が指数-対数関数の推定である、第8項記載の方法。

【0056】(11) a) 少なくとも1つの加算ツリーにおいて、確率メトリクスを組み合わせる工程と、

b) 別の可能性に対応する前記メトリクスの1つずつを組み合わせるように、最大値を見出す演算を実行する工程と、

c) 前記工程b)の結果に対する正規化演算を実行する工程とを備え、前記工程a)、b)およびc)を互いに平行化されたパイプライン状態で少なくとも部分的に実行する、平行MAP処理方法。

【0057】(12) 最大値を見出す演算が指数-対数式である、第11項記載の方法。

【0058】(13) 最大値を見出す演算が指数-対数関数の推定である、第11項記載の方法。

【0059】(14) スライディングウィンドウブロックに分割されたデータストリームをMAP処理するため

13

のシステムであって、アルファ発生プロセスと、ベータ発生プロセスとを備え、アルファプロローグおよびベータプロローグを使用して多数のスライディングウィンドウブロックに対する演算を行うように、アルファ発生プロセスとベータ発生プロセスとが多数のパイプライン化ステージに分割されている、MAP処理システム。

【0060】(15) スライディングブロックのMAPデコーダが前方伝搬計算および後方伝搬計算をパイプライン化するターボデコーダである。

【0061】

【図面の簡単な説明】

【図1】ターボデコーダのブロック図である。

【図2】パラレルスライディングウィンドウ処理を使用するMAPデコーダのブロック図である。

【図3】MAPデコーダ内のデータ発生ブロックのブロック図である。

【図4】MAPデコーダ内のアルファ発生ブロックのブロック図である。

14

【図5】MAPデコーダ内の外因性データ発生ブロックのブロック図である。

【図6】ベータブロック内のパイプライン化のタイミング図である。

【図7】アルファスライディングウィンドウブロックの発生とベータスライディングウィンドウブロックの発生との間のタイミングのずれを示す。

【図8】プロローグと共にアルファスライディングウィンドウブロックとベータスライディングウィンドウブロックとの間の対応を示す。

【図9】ベータビットおよびアルファビットを処理する順序の一例を示す。

【符号の説明】

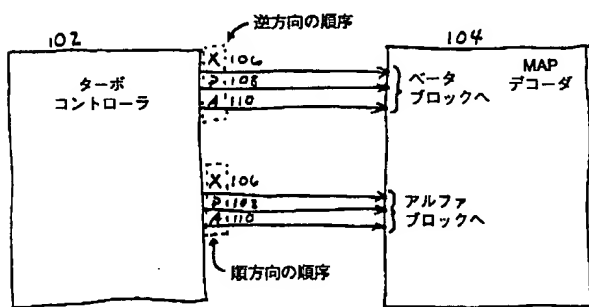
202 ベータブロック

204 ベータRAM

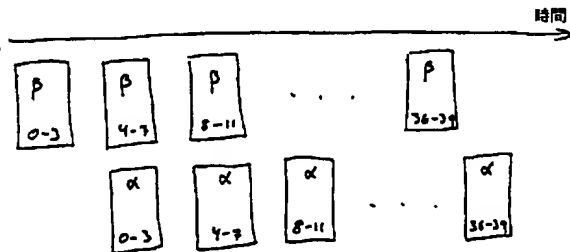
206 アルファブロック

212 外因性ブロック

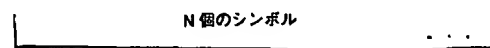
【図1】



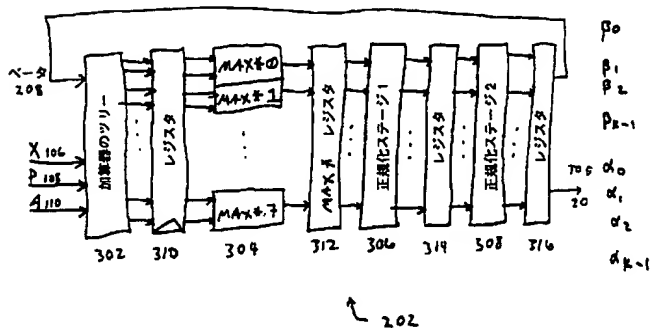
【図7】



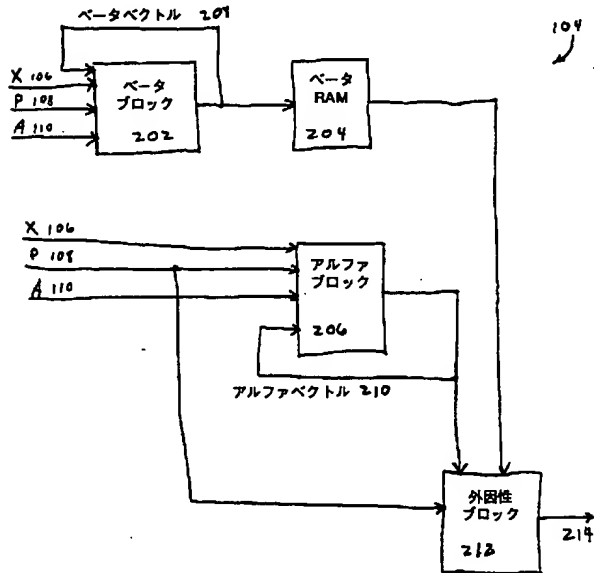
【図8】



【図3】



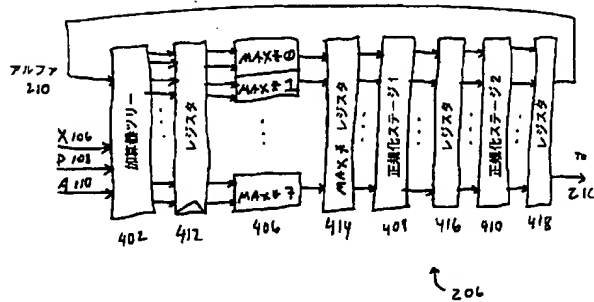
【図2】



【図9】

	スタート プロローグ	終了 プロローグ	スタートの 信頼性	終了の 信頼性
β_0	123	100	99	0
β_1	223	200	199	100
β_2	323	300	299	200
β_3	423	400	399	300
α_0	N/A	N/A	0	99
α_1	76	99	100	199
α_2	176	199	200	299
α_3	276	299	300	399

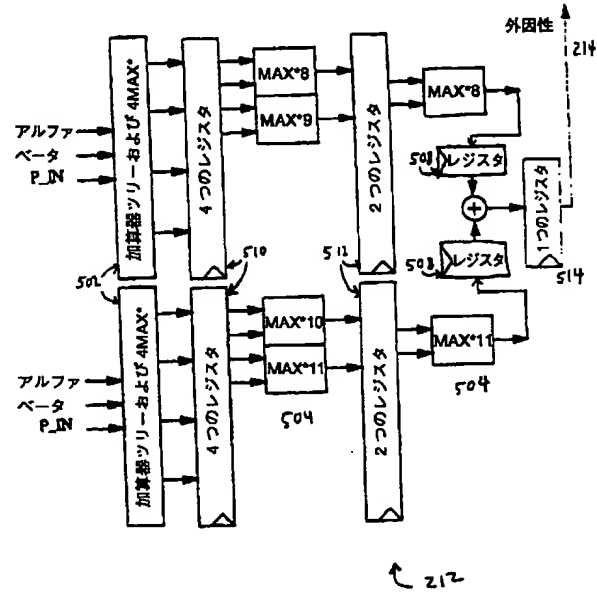
【図4】



【図6】

クロック サイクル	1	2	3	4	5	時間
β_0, α_0	β_0, α_0	β_0, α_1	β_0, α_2	β_0, α_3	β_0, α_4	
	β_1, α_0	β_1, α_1	β_1, α_2	β_1, α_3	β_1, α_4	
		β_2, α_0	β_2, α_1	β_2, α_2	β_2, α_3	
			β_3, α_0	β_3, α_1	β_3, α_2	
				β_4, α_0	β_4, α_1	
					β_5, α_0	

【図5】



フロントページの続き

(72)発明者 アルメル レイヌ
 フランス国 サンタンティープ、 リュ
 オーベルノン 13